

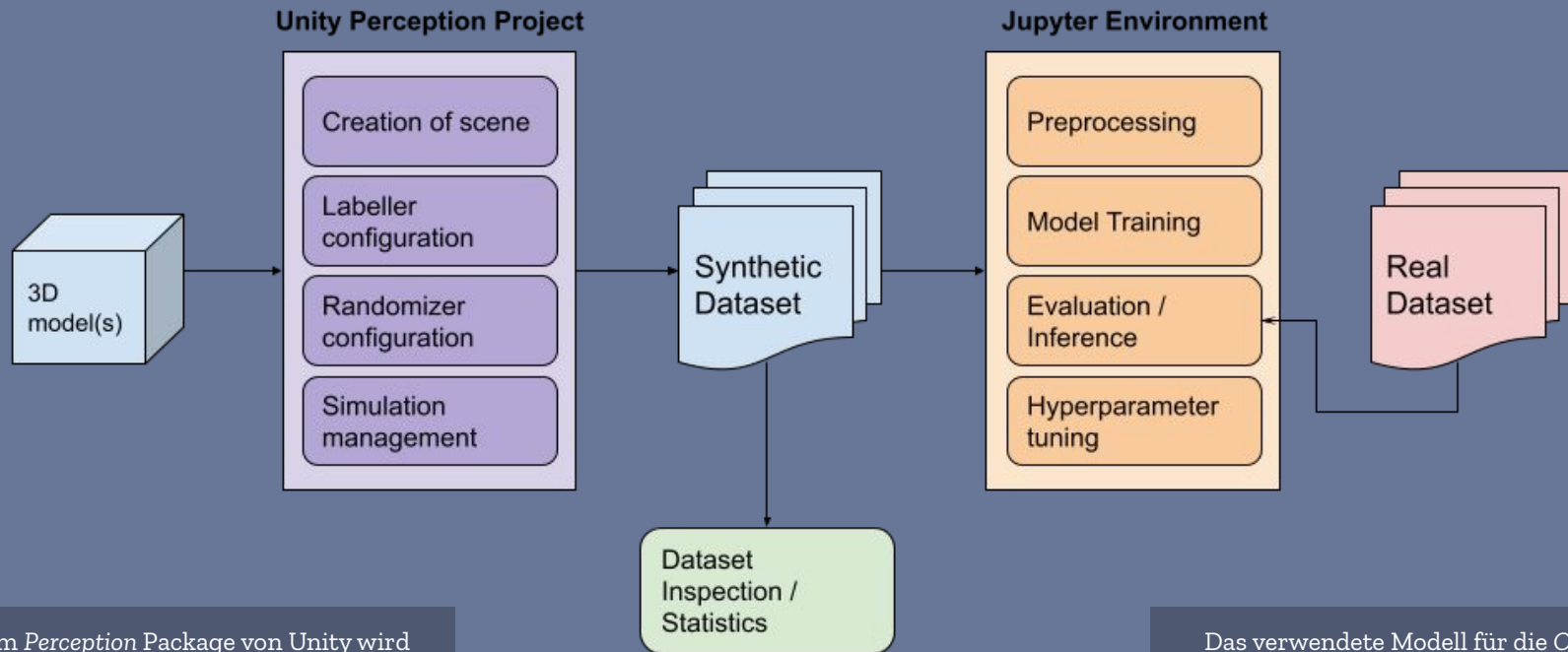
Im Rahmen meiner Masterthesis mit dem Arbeitstitel "*Synthetic Data with Unity for ML-assisted Pose Estimation*" befasse ich mich mit der zugrundeliegenden Problematik, dass für das Training von robusten ML-Modellen sehr viele Daten benötigt werden.

Mit einer physics-based Engine wie Unity, lässt sich eine Vielzahl von automatisiert gelabelten Bildern im Handumdrehen generieren.

Neben dem Aufbau einer entsprechenden Pipeline, gehe ich u.A. der Frage nach, wie die *Domain Gap* minimiert werden kann.

UNITY MEETS MACHINE LEARNING

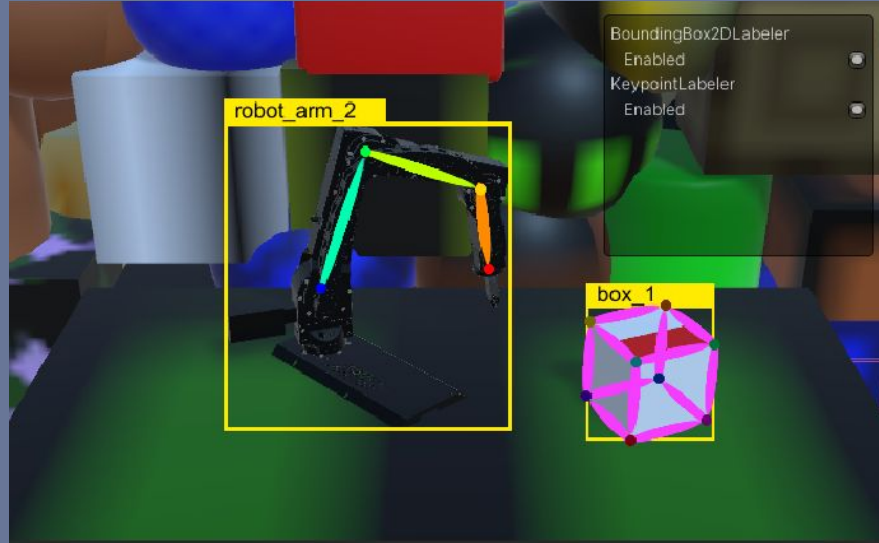
Pipeline der praktischen Umsetzung



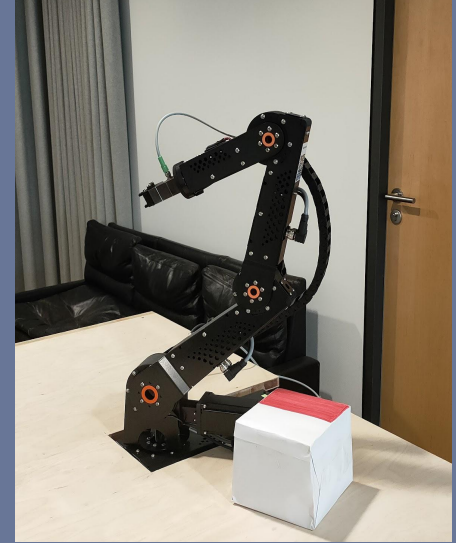
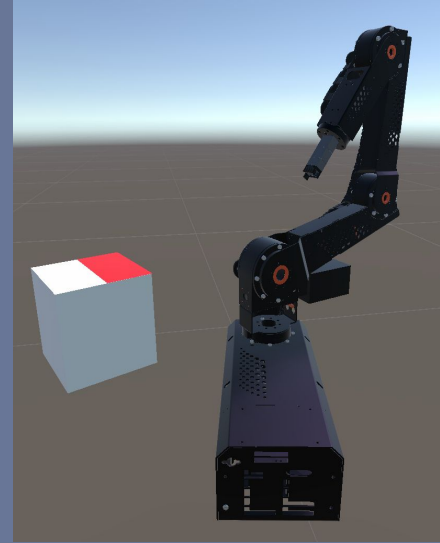
Mit dem *Perception Package* von Unity wird die Engine zur Erstellung von synthetischen Daten optimiert.

Das verwendete Modell für die *Object- und Keypoint Detection* ist ein auf dem COCO-Datensatz vortrainiertes Keypoint R-CNN.

Synthetische und echte Daten

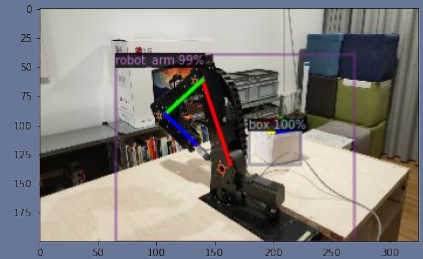
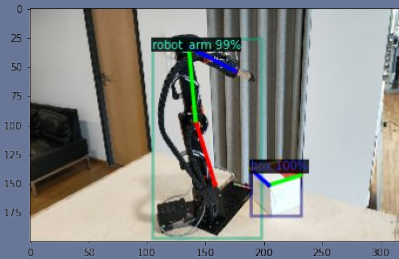
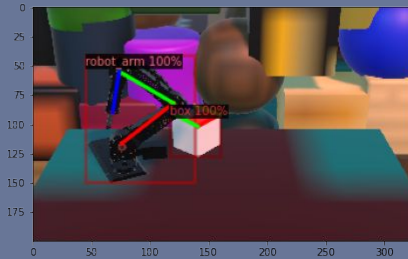
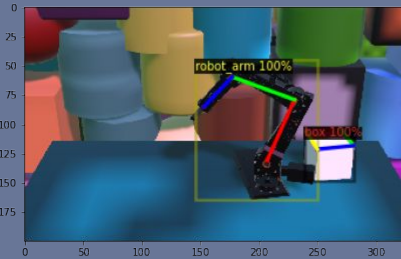
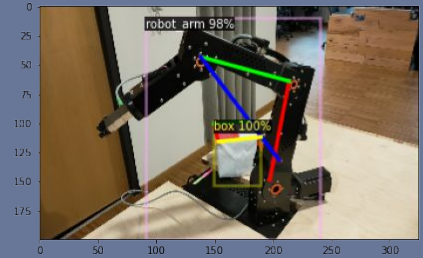
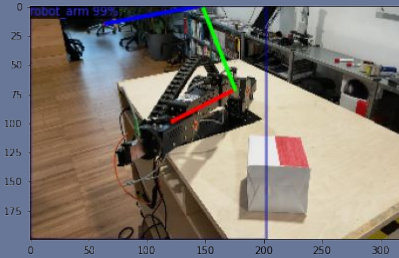
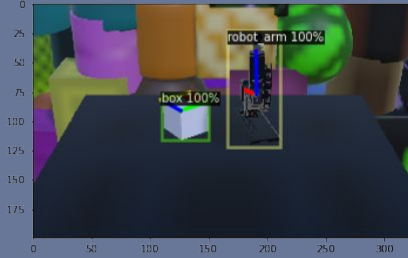
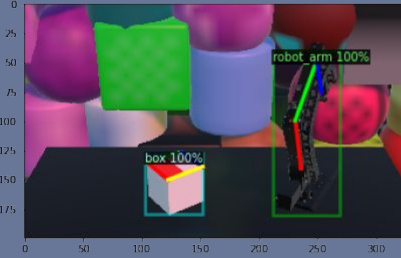


Die Daten werden nach dem Prinzip der *Domain Randomization* generiert. Zahlreiche Randomizer und Distractor-Elemente im Hintergrund zwingen das ML-Modell, besser zu generalisieren.



Direkter Vergleich zwischen dem echten Roboterarm und Box vs. deren digitale Zwillinge. Die Erkennung der Objekte an sich und der jeweiligen Keypoints soll die Basis für ein *collision avoidance system* bilden.

Predictions des trainierten Modells



Das auf synthetischen Daten trainierte Modell funktioniert aktuell bereits relativ gut auf echten Testdaten, hat aber noch Luft nach oben, speziell beim Zeichnen der *bounding box*.